2019-12-10 kl. 17-21

Tentaupplägg

TIPS 1:

Läs igenom ALLA uppgifterna. Välj den du känner är lättast först. Det kan gärna ta 10-20 minuter. Försök skriva saker som kan vara problem i uppgifterna. Är det något du absolut kommer att fastna på så kanske det är fel uppgift att ge sig på. Tiden du lägger på att noga läsa uppgifterna tjänar du in på att välja rätt uppgift.

TIPS 2:

Kolla ibland till kommunikationsfönstret. Det kan ha kommit information till alla utan att ni skickat in en fråga. Kanske gäller det dig också (d.v.s. den uppgift du jobbar med).

TIPS 3:

Om ni har problem med kompilator, Emacs eller annat som INTE har med uppgifterna att göra, räck upp handen så kommer en assistent. Detsamma gäller om hur man kopierar givna filer " cp given_files/* . " eller liknande.

Frågor om själva uppgifterna tar vi i första hand via tentasystemet.

I körexemplen har vi markerat det som användaren matar in på tangentbordet med ***fet och kursiverad*** stil. Tänk på att körexemplen bara är *ett* exempel på när programmet körs. Testa ditt program noga och tänka över hur programmet skall fungera vid andra indata.

Vi hinner normalt sett inte svara på frågor de sista 10 minuterna på tentan. Då ägnar vi all tid åt att rätta uppgifter för att alla skall hinna få svar innan ni går hem.

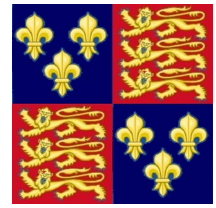
OBS: Tänk på att uppgifterna är numrerade 5-8. Det är viktigt att du skickar in din lösning på rätt uppgiftsnummer.

Betygsgränser:	**Tid**	**TekFak**
1 poäng	21:00	Betyg 3
2 poäng	20:30	Betyg 4
2 poäng	19:00	Betyg 5
>=3 poäng	20:00	Betyg 5

Bonustid från laborationer tillkommer till dessa tidsgränser. Tiden för betyg 3/G överstiger dock inte fyra timmar.

Lycka till med tenterandet och hoppas att alla får G på minst en uppgift idag.
M.v.h.
/Torbjörn (examinator)

Uppgift 5 – Tudorflaggan [1p]



Flaggan till höger är Englands kungliga flagga under Henrik VIII. Skriv ett program som låter användaren mata in önskad storlek och som matar ut en sådana flagga enligt nedanstående körexempel. Programmet skall fungera för godtyckliga positiva heltal.

Körexempel 1:

```
Mata in storlek: 1
+-------+
|^.^|\\\|
|---|---|
|\\\|^.^|
+-------+
```

Körexempel 2:

```
Mata in storlek: 2
+-----------+
|^.^|\\\|^.^|
|---|---|---|
|\\\|^.^|\\\|
|---|---|---|
|^.^|\\\|^.^|
+-----------+
```

Körexempel 3:

```
Mata in storlek: 3
+---------------+
|^.^|\\\|^.^|\\\|
|---|---|---|---|
|\\\|^.^|\\\|^.^|
|---|---|---|---|
|^.^|\\\|^.^|\\\|
|---|---|---|---|
|\\\|^.^|\\\|^.^|
+---------------+
```

Körexempel 4:

```
Mata in storlek: 10
+-------------------------------------------+
|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|
|---|---|---|---|---|---|---|---|---|---|---|
|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|
|---|---|---|---|---|---|---|---|---|---|---|
|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|
|---|---|---|---|---|---|---|---|---|---|---|
|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|
|---|---|---|---|---|---|---|---|---|---|---|
|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|
|---|---|---|---|---|---|---|---|---|---|---|
|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|
|---|---|---|---|---|---|---|---|---|---|---|
|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|
|---|---|---|---|---|---|---|---|---|---|---|
|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|
|---|---|---|---|---|---|---|---|---|---|---|
|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|
|---|---|---|---|---|---|---|---|---|---|---|
|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|
|---|---|---|---|---|---|---|---|---|---|---|
|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|\\\|^.^|
+-------------------------------------------+
```

KRAV: Din kod skall vara väl uppdelad i funktioner. Inget huvud/underprogram skall ha mer än 15 rader kod.

2019-12-10 kl. 17-21

Uppgift 6 – Explorer_Type [1p]

Fjorton- och femtonhundratalet var verkligen en tid för upptäcktsresande! Du skall i denna uppgift simulera upptäcktsresor för de relevanta stormakterna Spanien, Frankrike och England.

På filen EXPLORERS.TXT finns det data som representerar upptäcktsresande personer. Varje sådan har följande delar:

- Ett namn (en sträng).
- En nationalitet (en sträng, ”Spanien”, ”Frankrike” eller ”England”).
- Ett färdighetsvärde, (heltal mellan -1 och 4, hur pass duktig personen var som upptäcktsresande).

Du skall skapa ett program som definierar en Struct Explorer_Type som kan lagra datat för **en** upptäcktsresande. Programmet skall sedan lagra **alla** upptäcktsresande i en Vector av Explorer_Type. Du behöver inte låta programmet läsa in från filen utan datat kan hårdkodas in direkt i programmet. Programmet skall sedan slumpa fram en upptäcktsresande och presentera datat för användaren.

Körexempel 1:

```
Frankrike sänder ut Roberval! (0 färdighet)
```

Körexempel 2:

```
Spanien sänder ut Magellan! (4 färdighet)
```

2019-12-10 kl. 17-21

Uppgift 7 – Eltimer [2p]

Den som inte vill komma hem till ett mörkt hem kan köpa sig en timer. Det är en apparat som man sätter i vägguttaget och sedan kopplar man in sin lampa i den. På timern kan man ställa in exakt vilka tider man vill att lampan skall lysa.

Skapa ett program som sköter hanteringen av klockslag för timern. Man skall kunna välja exakt vilka sekunder under ett dygn som timern ska få lampan att lysa, alltså mellan 00:00:00 och 23:59:59. Vi antar att användaren alltid matar in korrekta klockslag. Huvudprogrammet skall låta användaren mata in klockslag. Om användaren skulle ange ett klockslag som redan finns i timern, så skall ett felmeddelande skrivas ut. Programmet avslutas genom att användaren trycker enter och ctrl-d.

KRAV 1: Du skall ha en struct Time_Type i ditt program som innehåller tre heltal. Använd denna för att lagra klockslag. Du skall ha en funktion ***get*** som läser in data till **en** variabel av typen Time_Type. Datatypen skall definieras på en separat headerfil *time_handling.h* med en tillhörande implementationsfil *time_handling.cpp*.

KRAV 2: I huvudprogrammet skall du använda vector på något sätt.

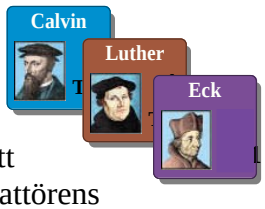
Körexempel:

```
Mata in klockslag: 23:15:45
Tiden registrerad i timern.
Mata in klockslag: 03:03:10
Tiden registrerad i timern.
Mata in klockslag: 23:15:45
Fel! Detta klockslag är redan registrerat!
Mata in klockslag: 16:09:00
Tiden registrerad i timern.
Mata in klockslag:
[ctrl-d]
```

OBS: Eftersom programmet slumpar vid tärningsslagen så skall det (troligtvis) bli olika utgång varje gång man kör programmet. Det betyder även att ditt programs utdata kommer skilja mot ovanstående körexempel.

2019-12-10 kl. 17-21

Uppgift 8 – Debattörer från förr [2p]



På femtonhundratalet skedde det mången teologisk debatt efter att Martin Luther dragit igång reformationen. Du skall i denna uppgift simulera vardagen för dessa katolska och protestantiska talare.

På filen DEBATERS.TXT finns många teologiska debattörer. På varje rad finns ett namn, religiös tillhörighet (’K’ markerar katolik, ’P’ markerar protestant) och debattörens färdighetsvärde (används inte i denna uppgift). Skriv ett program som först läser igenom filen och räknar hur många rader den har, och skriver ut detta. Därefter skall programmet göra följande **två gånger**:

1. Slumpa ett tal mellan 1 och antalet rader, ta motsvarande rad från filen.
2. Skriva ut namnet på debattören, och den religiösa tillhörigheten.

Om det blev olika religiös tillhörighet på debattarörna så blir det en vild debatt! I övriga fall överlägger debattörerna inför nästa debatt. Vi tillåter att samma debatör förekommer två gånger.

Körexempel 1:

```
Det finns 29 debattörer i filen DEBATERS.TXT.
Idag följer vi protestanten Olivetan och protestanten Knox.
De överlägger inför nästa teologiska debatt.
```

Körexempel 2:

```
Det finns 29 debattörer i filen DEBATERS.TXT.
Idag följer vi katoliken Campeggio och katoliken Faber.
De överlägger inför nästa teologiska debatt.
```

Körexempel 3 (antag att vi har tagit bort två rader ur filen):

```
Det finns 27 debattörer i filen DEBATERS.TXT.
Idag följer vi katoliken Eck och Protestanten Cop.
Det blir en vild debatt!
```

KRAV: Du får **inte** lagra hela innehållet i textfilerna i datorns minne. Läs lite i taget.

OBS: Eftersom programmet slumpar så skall det (troligtvis) bli olika utgång varje gång man kör programmet. Det betyder även att ditt programs utdata kommer skilja mot ovanstående körexempel.